Physical Data Organisation

Topics:

• Storage hierarchy
• Storage structures
• ISAM
• B-Trees
• Hashing
• Clustering
Storage Hierarchy

1 K (Kilo) = 10^3
1 M (Mega) = 10^6
1 G (Giga) = 10^9
1 T (Tera) = 10^{12}
1 P (Peta) = 10^{15}

8 - 512 Byte/Register

Compiler

~10 MB Byte/Cache

cache-controller

~100 GB-range, 64B block size

operating system

~1 TB-range

4kB blocks | user

PB-range

gerator user
Storage Hierarchy

1 n (nano) = $10^{-9}$
1 μ (micro) = $10^{-6}$
1 m (milli) = $10^{-3}$

< 1ns
< 10 ns
< 100 ns
< 10 ms
secs

(Flash-Memory Lower TB-range, < 100 μs)

Register

L1/L2/L3 Cache

Main Memory

Disk

Tape
Storage Hierarchy

- Room (1min)
- Building (10min)
- City (1.5h)
- Mars (2 months)
- Pluto (9 years)
- Andromeda (2000 years)

- Register: < 1ns
- L1/L2/L3 Cache: < 10ns
- Main Memory: < 100ns
- Disk: < 10 ms
- Tape: secs
- Storage Hierarchy
Storage Hierarchy

Room (1min)  
- < 1 ns register

Building (10min)  
- < 10 ns L1/L2/L3 Cache

City (1.5h)  
- < 100 ns Main Memory

(Mars (2 month))  
- < 10 ms Disk

Pluto (9 years)  
- secs

Andromeda (2000 years)  
- Tape

Factor 100.000

Mars (2 months)
Buffer Management

Main Memory

Disk
Buffer Management

- Main Memory
- Disk
- replace
- fill

disk ~ persistent DB
Fill and replace pages

- System buffer is divided in frames of equal size
- A frame can be filled with one page
- Overflow pages are swapped on disk

<table>
<thead>
<tr>
<th>Main Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>16K</td>
</tr>
<tr>
<td>32K</td>
</tr>
<tr>
<td>48K</td>
</tr>
</tbody>
</table>

Diagram:
- Disk (swap device)
- Main Memory
- (Buffer) Frames
- Page
- Frames
Addressing tuples on disk

Page 4711

TID
4711 2

1 2 3

5001 ◦ Grundzüge ◦ ...
4052 ◦ Logik ◦ ...
5041 ◦ Ethik ◦ ...
Moving within a page

TID
4711 2

1 2 3

5001  Grundzüge  

5041  Ethik  

4052  Mathematische  Logik  

Page 4711
Moving from one page to another

TID
4711 2

5001  Grundzüge ... 5041  Ethik ...

4812 3

4052  Mathematische Logik für Informatiker ...

Page 4711

Page 4812

Forward
Moving from one page to another

With the next move the „Forward“ on page 4711 is altered (no more Forward to page 4812)
Storage Summary

Storage Hierarchy
- Huge/slow storage vs small/fast memory
- Very important for DBMSs design
- Algorithms need to be aware of performance difference and place data optimally

Buffer Management
- A component of the database
- Migrates pages between disc and main memory
- Keeps hot pages in DRAM and cold ones on disc

Tuple IDs
- Used to locate a tuple
- Composed of page identifier and a page-local record identifier
Storage Operations

Full table scan: Retrieve all tuples from a table
\[
\text{select} \ * \ \text{from} \ \text{students};
\]

Point query: Find one specific tuple
\[
\text{select} \ * \ \text{from} \ \text{students}
\quad \text{where} \ \text{studNr} = 26120;
\]

Range query: Find one specific tuple
\[
\text{select} \ * \ \text{from} \ \text{students}
\quad \text{where} \ 26000 \leq \text{studNr} \quad \text{and} \ \text{studNr} < 27000;
\]
Index Motivation

**Full table scan**: Load all pages from disk, one by one.

**Point query**: When we sort the data, we can find a key more easily. Compare to a dictionary: You can look at the page in the middle to determine in which half the word you are looking for is and then continue this process with the new found half. This way you need to look at much less pages then reading it front to back.

**Range query**: When the data is stored in a sorted fashion, a range query can be processed as a combination of a point query (to find the starting point) and then a scan.
Index Motivation

Page 1
24002 Xenokrate
25403 Jonas
26120 Fichte

Page 2
26830 Aristoxenos
2755 Schopenhauer

Page 3
28106 Carnap
29120 Theophrastos
29555 Feuerbach

RAM
Disk
Data Transfer

Simple query execution:
**select * from students where studNr=26120;**

Get one tuple/page after the other to the main memory and evaluate predicates.

→ Most expensive way 😞
→ Mostly only a small fraction of the tuples fulfills the query
Index Structures

- Index structures are used to keep the data volume to be transferred from disk to main memory small.
- Only that part of the data which is really needed to answer the query is transferred.
- Two main indexing methods:
  - Hierarchical (trees)
  - Partitioning (Hashing)